The Minimal Cut Cover of a Graph

Michael Bussieck *

March 9, 1994

Abstract

We consider a problem which occurs in testing for short circuits in printed circuit board components. This problem can be modeled as the covering of the edges of an undirected graph by cuts.

In particular, we look for a family of cuts $\mathcal{F} = \{(C_1, \overline{C_1}) \ldots, (C_\Phi, \overline{C_\Phi})\}$ with minimal cardinality Φ, such that each edge belongs to at least one $(C_i, \overline{C_i})$. We call this problem the *Minimal Cut Cover* problem, MCC for short. For the special case of complete graphs, Loulou [7] describes a polynomial greedy-like method for MCC.

Here we will show that MCC is NP-hard for arbitrary graphs, by drawing the relationship to the vertex color problem. Furthermore we give some algorithmic approaches for solving MCC approximately.

1 Introduction

In this paper we will discuss a problem arising in testing printed circuit boards (PCB's). We give a brief description of the real-world problem, and refer to [7] for a detailed overview. The PCB's have to stand some electronic tests. The *in-circuit* test consists of a battery of tests performed on the (interconnected) components via an automatic, programmable testing machine. One of the test is the so called *'shorts' test*, i.e. testing for short circuits of components which proceed as follows: for a given PCB the tester's fixture is equipped with probes which can be connected to a polarized electric source (either "+" or "–"). The probes are lowered onto the connecting points of the PCB, all at once. This turn tests all components with one end connected to the "+" source and the other to the "–" source. This procedure must be repeated until all components are checked in this way.

Loulou [7] suggests a representation of the PCB by a graph $G = (V, E)$ in which each vertex represents a connecting point (area of same potential) and each edge represents one component (cf. figure 1). Each assignment of the connecting points to "+" and "–" sources corresponds to a *cut* in G, i.e. a partition $(T, \overline{T})$ of V, which covers (checks) the edges (components) with one vertex in T and the other vertex not in T. In order to minimize the time and costs of the testing phase we look for a minimum cardinality family of cuts which covers all edges $e \in E$. We call this problem the *minimal cut cover problem* (MCC for short). Let $\Phi(G)$ denote the size of a MCC of G.

The rest of the paper is organized as follows. In section 2 we describe the relationship between MCC and the *vertex color problem* (VC for short) and show that MCC is NP-hard for arbitrary graphs. In section 3 we consider some algorithmic approaches to MCC, suggest and analyze some simple heuristics. In section 4 we give some concluding remarks.

2 Relationship between cut-cover and vertex-color

Let $G = (V, E)$ be an undirected graph. $f: V \to \boldsymbol{N_0}$ is called a *valid vertex coloring* if $f(u) \neq f(v)$ for every $uv = e \in E$. The minimal number $|f(V)| = \chi$ of colors of a valid vertex coloring f is called the *chromatic number*.

*TU Braunschweig, Abteilung für Mathematische Optimierung, Pockelsstraße 14, D-38106 Braunschweig, Germany

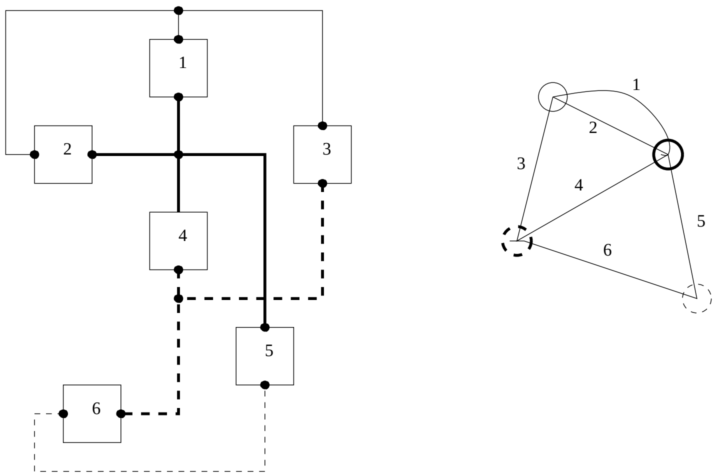

Figure 1: A PCB and its representation

Theorem 1 *Let $G = (V, E)$ be an undirected graph. G has a valid vertex coloring $f: V \longrightarrow \boldsymbol{N_0}$ with at most $k \leq 2^q$ colors, if and only if G has a cut-cover $\mathcal{F} = \{(C_1, \overline{C_1}), (C_2, \overline{C_2}), \ldots, (C_q, \overline{C_q})\}$ of size q.*

Proof. Suppose G has a cut-cover $\mathcal{F} = \{(C_1, \overline{C_1}), (C_2, \overline{C_2}), \ldots, (C_q, \overline{C_q})\}$ of size q. We define a function $\tilde{f}: V \longrightarrow \{0, 1\}^q$ by

$$\tilde{f}(v)_i := \begin{cases} 1 & v \in C_i \\ 0 & v \notin C_i \end{cases} .$$

We can interpret $\tilde{f}(v)$ as a binary coded number. So we define $f(v) := \sum_{i=1}^{q} 2^{i-1} \tilde{f}(v)_i$. Consider an edge $uv = e \in E$. Let $(C_i, \overline{C_i})$ be the cut which covers e. Since $\tilde{f}(v)_i \neq \tilde{f}(u)_i$ we have $f(v) \neq f(u)$ and hence f is a valid vertex coloring.

Now consider a valid vertex coloring $f: V \longrightarrow \boldsymbol{N_0}$ with $k \leq 2^q$ colors. Without loss of generality suppose $f(V) = \{0, 1, \ldots, k-1\}$. Let $\tilde{f}: V \longrightarrow \{0, 1\}^q$ be the binary representation of f. We define a family $\mathcal{F} = \{(C_1, \overline{C_1}), (C_2, \overline{C_2}), \ldots, (C_q, \overline{C_q})\}$ of size q by $C_i := \{v \in V | \tilde{f}(v)_i = 1\}$. Consider an edge $uv = e \in E$. Since f is a valid coloring we have $f(u) \neq f(v)$. There must be an index i_0 for which $\tilde{f}(u)_{i_0} \neq \tilde{f}(v)_{i_0}$ holds. Hence e is covered by $(C_{i_0}, \overline{C_{i_0}})$ and $\mathcal{F}$ is a cut-cover. □

Remark 1 *Providing a coloring f with k colors, the proof of theorem 1 implies an easy $O(n \cdot \log k)$ algorithm for constructing a cut cover and vice versa.*

The next lemma shows how we can obtain from a MCC a near optimal VC.

Lemma 1 *Let $\mathcal{F} = \{(C_1, \overline{C_1}), (C_2, \overline{C_2}), \ldots, (C_k, \overline{C_k})\}$ be a* MCC *of an undirected graph G. Then we have $\frac{2^k - \chi(G)}{\chi(G)} < 1$.*

Proof. Suppose the opposite $\frac{2^{\Phi} - \chi(G)}{\chi(G)} \geq 1$ holds. Then we have $2^{\Phi - 1} \geq \chi$. By theorem 1 we can find a cut-cover of size $\Phi - 1$. This contradicts the minimality of $\mathcal{F}$. □

Let A be an approximation algorithm for a minimization problem π. $A(G)$ $(\pi(G))$ denotes the value of the solution found by A (optimal solution).

Corollary 1 MCC *is a NP-hard problem.*

Proof. The result is obvious by lemma 1 and an early result of Garey and Johnson [4]. They prooved that until P$\neq$NP there is no polynomial time approximation algorithm for VC with *performance ratio* $\frac{A(G)-\chi(G)}{\chi(G)} < 1$. □

Since no optimal polynomial time algorithm is expected, we can turn our attention to suboptimal algorithms. Let A be an approximation algorithm for a minimization problem π. Similar to performance ratio we define the *performance difference* by $A(G) - \pi(G)$.

The relation of optimal solutions for VC and MCC worked out in theorem 1 continues for approximate solutions as lemma 2 shows.

Lemma 2 1. *If an algorithm* A_{VC} *for* VC *guarantees a performance ratio less than* ϵ*, then there is an algorithm* A_{CC} *for* MCC *which guarantees* $A_{CC}(G) - \Phi < \log_2(\epsilon + 1)$.

2. *If an algorithm* A_{CC} *for* MCC *guarantees a difference performance less than* ϵ'*, then there is an algorithm* A_{VC} *for* VC *which guarantees* $\frac{A_{VC}(G)-\chi}{\chi} < 2^{\epsilon'+1} - 1$.

Proof.

1. With algorithm A_{CC} mentioned in remark 1 we construct a cut-cover of cardinality $\lceil \log_2 A_{\text{VC}}(G) \rceil$. Then we have

$$A_{\text{CC}}(G) - \Phi = \lceil \log_2 A_{\text{VC}}(G) \rceil - \lceil \log_2 \chi \rceil \leq \left\lceil \log_2 \frac{A_{\text{VC}}(G)}{\chi} \right\rceil < \lceil \log_2(\epsilon + 1) \rceil.$$

2. Similar the case abouve we find a VC with $2^{A_{\text{CC}}(G)}$ using algorithm A_{VC} mentioned in remark 1. We obtain

$$A_{\text{CC}}(G) - \Phi < \epsilon' \Leftrightarrow 2^{A_{\text{CC}}(G)-\Phi} < 2^{\epsilon'} \Leftrightarrow \frac{A_{\text{VC}}(G)}{2^{\Phi}} < 2^{\epsilon'} \Rightarrow \frac{A_{\text{VC}}(G) - \chi}{\chi} < 2^{\epsilon'+1} - 1 \qquad □$$

3 Algorithmic Aspects

In section 1 we mentioned that the MCC problem is of practical relevance. The method used at some plants (cf. [7]) to determine a (valid) cut-cover is the following. Let $V = \{v_1, v_2, \ldots, v_n\}$ then $\mathcal{F} = \{(V \setminus v_1, v_1), (V \setminus v_2, v_2), \ldots, (V \setminus v_n, v_n)\}$ is a cut-cover of size n. In [7] Loulou gives a method that reduces the size of the cut-cover from n to $\lceil \log_2 n \rceil$ or less. The method is based on the following corollary.

Corollary 2 *Let* K_n *be the complete graph with* n *vertices. The* MCC *of* K_n *consists of* $\lceil \log_2 n \rceil$ *cuts.*

Proof. Since K_n can be colored with n colors, theorem 1 ensures that there is a cut-cover with $\lceil \log_2 n \rceil$ cuts. A cut-cover with $\lceil \log_2 n \rceil - 1$ cuts or less implies a coloring of K_n with less than n colors. This contradicts $\chi(K_n) = n$. □

The algorithm to find a MCC for complete graphs, suggested by Loulou, works in a greedy fashion. It determines a cut $(X, \overline{X})$, such that $\delta(X)$ is maximal, where $\delta(X)$ denotes the number of edges from X to $\overline{X}$. For general graphs the problem to find a cut with such max-property is NP-complete (cf. [5] MAX-CUT) but for complete graphs each set X with $|X| = \lceil \frac{n}{2} \rceil$ is a MAX-CUT. The Greedy algorithm works as follows. Take an arbitrary set X with $X = \lceil \frac{n}{2} \rceil$. After cancelling the edges $\delta(X)$ the graphs splits into two complete graphs of size $\lceil \frac{n}{2} \rceil$ and $\lfloor \frac{n}{2} \rfloor$. For these graphs we repeat the procedure until all edges are annihilated. It is easy to see that for complete graphs this algorithm generates the same solution as the algorithm mentioned in theorem 1 with a certain coloring.

We suggest a method that is related with the greedy-algorithm, i.e. in each iteration we try to annihilate a huge number of edges. Unfortunately even annihilating the edges of the MAX-CUT in

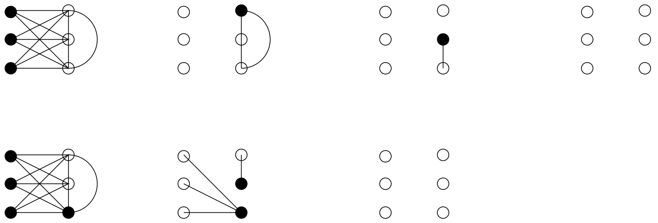

Figure 2: A counter example for the Greedy-Algorithm

each iteration will not lead to a MCC. See the example in figure 2. The upper sequence of graphs is produced by the greedy algorithm. The lower sequence is optimal.

We approximate the MAX-CUT by the edges of a spanning DFS-(depth-first-search)-forest. Let $T = (V, E_T)$ be a spanning DFS-forest of $G = (V, E)$ with roots $r_1, \ldots, r_p \in V$ of the disjoint trees in $T = (V, E_T)$. Let v a vertex in a tree with root r_i then $l(v)$ denotes the length of the path from r_i to v. ($l(r_i) = 0$). We define the cut $(X, \overline{X})$ taken to family $\mathcal{F}$ by $X := \{v \in V \mid l(v) \text{ is odd}\}$. We name this algorithm Tree-Cover.

Remark 2 *For complete graphs the* Tree-Cover *heuristic provides the optimal solution (i.e.* $A_{TC}(K_n) = \lceil \log_2 n \rceil$*), since the cut found by* Tree-Cover *divides* K_n *into* $K_{\lceil \frac{n}{2} \rceil}$ *and* $K_{\lfloor \frac{n}{2} \rfloor}$ *(cf. figure 3).*

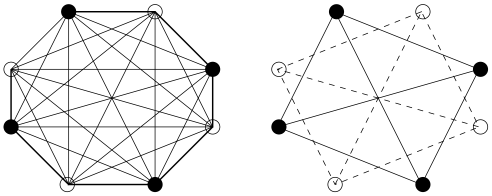

Figure 3: Iteration of Tree-Cover applied to K_n

Consider the complexity of Tree-Cover. Since Tree-Cover computes a DFS forest in each iteration we obtain a complexity of $O((m+n)\rho)$ where ρ denotes the number of iterations. Since we have for K_n only $\lceil \log_2 n \rceil$ iterations, the total complexity is $O((m+n)\log n)$.

Another approach makes use of the relation between the family of cuts and a special class of graphs. Let $\mathcal{F} = \{(C_1, \overline{C_1}) \ldots, (C_\Phi, \overline{C_\Phi})\}$ be the MCC of an undirected graph $G = (V, E)$. Define $G_\mathcal{F} = (V, E_\mathcal{F})$ with $E_\mathcal{F} := \{C_1 \times \overline{C_1} \cup \ldots \cup C_\Phi \times \overline{C_\Phi}\}$. $G_\mathcal{F}$ is a supergraph of G with $\Phi(G) = \Phi(G_\mathcal{F})$. $G_\mathcal{F}$ contains no path of length 4 without chords. Such graphs are called *cographs.* Cographs have been studied under a variety of names such as N-free graphs [2], hereditary Dacey graphs [10], reticles [9], and TSP digraphs [6]. Some intractable combinatorial optimization problems, like VC, can be solved in polynomial time using the unique representation tree, called *cotree.* In [1] Corneil et al. suggest a linear time recognition algorithm for cographs, which can we used to find a N-free supergraph $\tilde{G}$ of a general graph G. Since MCC can be efficiently solved for $\tilde{G}$, we approximate $\Phi(G)$ by $\Phi(\tilde{G})$.

Finally we mention an approach of great promise. Theorem 1 tells us that we can take every VC-heuristic (several sequential color heuristics, tabu search, etc. For an excellent overview cf. [3]) for MCC.

4 Concluding remarks

We have extended the results for complete graphs given in [7] to arbitrary undirected graphs. With the relationship between MCC and VC many results (NP-hardness, behavior of heuristics, etc) can be easily transferred from VC to MCC. From the practical point of view this relationship may be very useful. Turner [11] gives an algorithm of complexity $O(n + m \log k)$ that colors a k-colorable graph with k colors with probability close to 1 . Moreover when a PCB is designed we can spend some time to find an optimal or near optimal solution because this work has to be done only once. During the whole testing phase for a lot of same PCB's no further work must be done to configure the testing machine. With the relationship to VC we opened a rich source for heuristics approaches.

If the PCB imply a graph with "nice" properties, like planar, N-free, etc. then an MCC can be found in polynomial time.

The MCC problem has a natural extension, the k-MCC problem. Find a family $\mathcal{F}$of cuts with minimal cardinality such that each edge of an undirected graph $G = (V, E)$ is covered by at least k cuts of $\mathcal{F}$.

Even for complete graphs K_n the k-MCC problem is intractable. There is a related, well studied, problem in the theory of error correcting codes (cf. [8]). Find a code with n codewords and Hamming distance k under the objective to minimize the length of the codewords. Such a minimal code implies a k-MCC for K_n.

References

[1] D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM J. Comput.*, 14(4):926–934, 1985.

[2] Jacinta Covington. A universal structure for N–free graphs. *Proc. London Math. Soc. (3)*, 58:1–16, 1989.

[3] D. de Werra. Heuristics for graph coloring. *Computing Suppl.*, 7:191–208, 1990.

[4] M. R. Garey and D. S. Johnson. Approxiamtion algorithms for combinatorical problems: An annotated bibliography. In J. F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 41–52. Academic Press, New York, 1976.

[5] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.

[6] E. L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Ann. Disc. Math.*, 2:75–90, 1978.

[7] Richard Loulou. Minimal cut cover of a graph with an application to the testing of electronic boards. *Oper. Res. Lett.*, 12:301–305, 1992.

[8] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*, volume 16 of *North-Holland Mathematical Library.* North-Holland, 1977.

[9] J. H. Schmerl. Decidability and $\aleph_0$-categoricity of theories of partially orderd sets. *J. Symbolic Logic*, 45:585–611, 1980.

[10] D. P. Sumner. Dacey graphs. *J. Austral. Math. Soc. Ser. A*, 18:492–502, 1974.

[11] J. S. Turner. Almost all k-colorable graphs are easy to color. *J. Algorithms*, 9:63–82, 1988.